LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-680306

Literature review for vehicle correspondence and network modeling and analysis

K. Boakye, P. Kidwell, G. Konjevod, J. Lenderman

December 18, 2015

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Literature review for vehicle correspondence and network modeling and analysis

Kofi Boakye, Paul Kidwell, Goran Konjevod and Jason Lenderman

February 10, 2015

LLNL-TR-680306

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344

Chapter 1

A Literature Review for Vehicle Correspondence

1.1 Vehicle Recognition

The ability to recognize specific vehicle types (e.g., car make and model) is central to the correspondence task. Here we describe two recent efforts in this area.

1.1.1 "Attribute-based Vehicle Recognition using Viewpoint-aware Multiple Instance SVMs"[5]

The authors in this work present an approach for vehicle recognition that uses multiple instance learning to discover local attributes conditioned on viewpoint (e.g., side, front, back, etc.). To encourage discovered regions to be semantically meaningful, additional constraints on the positions of image regions across images are applied to the multiple-instance SVM (MI-SVM) model. Namely, for any pair of images within the same vehicle class, they assume that a good attribute should occur at similar places on the vehicle if the viewpoints of the two images are the same, or at similar places after an image transformation for different viewpoints. To determine viewpoints, an initial set of viewpoints labels is generated with K-means clustering using global image gradient features. These labels are then iteratively updated using a voting scheme which obtains votes from discovered regions in a given image according to the most likely viewpoint for each region. The most likely viewpoint is determined by the viewpoint of the detector that identified the region. To evaluate performance, the authors present results on the Stanford Cars and INRIA datasets. These results demonstrate that the approach does indeed identify semantically meaningful regions and that incorporating viewpoint information yields improved performance.

1.1.2 "Car Make and Model Recognition using 3D Curve Alignment"[11]

In this paper, fine-grained recognition of car make and model is achieved using 3D space curves obtained from multiple 2D views of cars. The multiple views are used to generate a 3D visual hull onto which the 2D image curves are backprojected and subsequently filtered for spurious edges. 3D chamfer matching is then used to generate multiple distance measures for the curves. For this

work, Euclidean distance and orientation distance are used. These two distances are then utilized as features for a logistic regression classifier for each make and model. Using an evaluation set of 190 images containing one of eight Honda models or 30 other random cars, the authors obtain a precision/recall area under the curve (AUC) of 0.799 for the 3D curve alignment system, compared 0.189 for a 2D keypoint baseline and 0.256 for a 2D chamfer one.

1.2 Person Re-Identification

Person re-identification, i.e., the task of recognizing a single person across spatially disjoint cameras, bears many similarities to vehicle correspondence. Consequently, the techniques and features employed in person re-identification have strong relevance and we give a brief overview of current approaches here.

1.2.1 "Viewpoint invariant pedestrian recognition with an ensemble of localized features"[8]

This paper introduces the ensemble of localized features (ELF) approach for pedestrian recognition and person re-identification. The proposed similarity function is a weighted ensemble of likelihood ratio tests constructed with the AdaBoost algorithm. The set of features are drawn from eight color channels and nineteen texture channels (from the Schmid and Gabor families of texture filters). Results on a pedestrian recognition dataset show significantly improved performance over various baseline histogram-based approaches.

1.2.2 "Person re-identification by support vector ranking"[10]

This paper converts the person re-identification task from one of absolute scoring to a relative ranking problem. That is, rather than focus on the classification of correct versus incorrect matches, the authors propose an approach based on the concept of document ranking, prevalent in information retrieval. Given a probe image for query, the objective is to find the most relevant gallery images arranged according to rank. In addition, the authors propose the Ensemble RankSVM, which builds on the standard RankSVM [3] and aims to overcome some of its scalability issues. Ensemble RankSVM uses an ensemble of weak RankSVMs, each computed on a small set of data, and then combines them using ensemble learning to build a stronger ranker. Performance results are presented on the VIPeR and a set of images from the i-LIDS dataset and demonstrate that (1) ranking-based approaches generally outperform non-ranking ones; and (2) the Ensemble RankSVM method is competitive with the state-of-the-art while being less computationally intensive.

1.2.3 "Person re-identification by symmetry-driven accumulation of local features"[6]

The authors propose a feature extraction and matching strategy, dubbed ıSymmetry-Driven Accumulation of Local Features (SDALF), that is based on features that model three complementary aspects of human appearance: (1) the overall chromatic content; (2) the spatial arrangement of colors into stable regions; and (3) the presence or recurrent local motifs with high entropy. The extraction proceeds by first isolating, the head, torso, and legs of the body. Then, for the last two parts, a vertical axis of appearance symmetry is estimated. Thereafter features are generated

and weighted by the distance to the vertical axis. This weighting seeks to minimize the effect of pose variations. The three features consist of (1) HSV histograms, (2) maximally stable color regions (MSCR) [7], and (3) recurrent highly structured patches (RHSP), which are estimated using a per-patch similarity analysis. Matches are computed based on weighted sum of the distances between the three feature types. The approach is evaluated on three datasets—VIPeR, i-LIDS, and ETHZ—and the authors demonstrate new state-of-the art performance for the first two datasets.

1.2.4 "Person re-identification by descriptive and discriminative classification"[9]

In this paper, the authors employ descriptive and discriminative appearance models in parallel to yield improvements in person re-identification capability. The descriptive model is derived from a descriptor that is obtained by computing the covariance between a set of visual features (which represent intensity, color, and texture) across pixel locations in a bounding box within the observed image. When a specific probe image is used as a query it is compared to all gallery images, which are then ranked according to a covariance-based similarity.

The discriminative model is used to refine the ranking result for those cases in which the true match has low rank. This information could potentially be provided by an analyst when the system is in operational mode. The model is estimated using a boosting approach, as in [12], with Haar and covariance features. The Haar features mainly capture intensity changes between the upper and lower body of a person while the covariance features extract local color information. The weak learner for the Haar features consisted of a Bayesian decision criterion while a multi-dimensional nearest neighbor classifer was used for the covariance features. The authors present results that demonstrate that the parallel approach can yield improved performance over either approach in isolation and that this approach is competitive with the state-of-the art while not requiring hand labeling or foreground-background segmentation.

1.3 Image Classification Features

A major component of vehicle correspondence is the development of appearance models for the vehicles of interest. For image classification approaches to correspondence this typically consists of extracting features from exemplar images and using those features in a classifier such as a random forest or support vector machine. While there are a number of features common to image classification, such as SIFT and HOG, new features continue to be investigated that may yield additional performance improvements. We discuss a few of these recently developed features below.

1.3.1 "Brownian descriptor: a Rich Meta-Feature for Appearance Matching"[2]

In this paper the authors introduce an image region descriptor inspired by recent studies in mathematical statistics related to Brownian motion. This Brownian descriptor, according to the authors, is a natural extension of covariance and measures non-linear relationships between features in contrast to the linear relationships measured by covariance. The Brownian descriptor introduces a new

covariance, the *distance covariance* $\mathcal{V}_n^2$ given by

$$\mathcal{V}_n^2(X,Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}B_{kl} \tag{1.1}$$

where A_{kl} and B_{kl} are simple linear functions of the pairwise distances between sample elements. The descriptor is applied to two vision tasks—visual tracking and person re-identification—and results show superior performance over standard covariance.

1.3.2 "COLOR CHILD: A robust and computationally efficient Color Image Local Descriptor"[1]

The Color moments augmented Cumulative Histogram-based Image Local Descriptor, or COLOR Child, is a dense image descriptor based on the Weber Law Descriptor (WLD) of Chen et al. [4]. This descriptor and its related variants are based on two components, namely differential excitation and orientation. COLOR Child uses differential excitation component based on the Laplacian of Gaussian (LoG) operator and orientation based on Tiansi fractional derviative filter [13]. These values are computed per pixel and then binned into histograms. For histogram comparison, the authors, propose the use of the Wasserstein distance, given by

$$W(F,G) = \int_{\mathbb{R}} |F(x) - G(x)|\ dx \tag{1.2}$$

where F and G are, respectively, the cumulative histograms of $H(I_1)$ and $H(I_2)$, the histograms of the two images. Performance results for texture classification who COLOR Child to outperform other Weber law based descriptors and a few non-Weber law based ones such as LBP, SIFT, LQP, GLCM and MLEP.

REFERENCES

[1] Sai Hareesh Anamandra and V Chandrasekaran. Color child: A robust and computationally efficient color image local descriptor. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 227–234. IEEE, 2014.

[2] Sławomir Bak, Ratnesh Kumar, François Bremond, et al. Brownian descriptor: a rich meta-feature for appearance matching. In *WACV: Winter Conference on Applications of Computer Vision*, 2013.

[3] Olivier Chapelle and S Sathiya Keerthi. Efficient algorithms for ranking with svms. *Information Retrieval*, 13(3):201–215, 2010.

[4] Jie Chen, Shiguang Shan, Chu He, Guoying Zhao, Matti Pietikainen, Xilin Chen, and Wen Gao. Wld: A robust local image descriptor. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1705–1720, 2010.

[5] Kun Duan, Luca Marchesotti, and David J Crandall. Attribute-based vehicle recognition using viewpoint-aware multiple instance svms.

[6] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367. IEEE, 2010.

[7] P-E Forssén. Maximally stable colour regions for recognition and matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[8] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Computer Vision–ECCV 2008*, pages 262–275. Springer, 2008.

[9] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Image Analysis*, pages 91–102. Springer, 2011.

[10] Bryan Prosser, Wei-Shi Zheng, Shaogang Gong, Tao Xiang, and Q Mary. Person re-identification by support vector ranking. In *BMVC*, volume 1, page 5, 2010.

[11] Krishnan Ramnath, Sudipta N Sinha, Richard Szeliski, and Edward Hsiao. Car make and model recognition using 3d curve alignment.

[12] Kinh Tieu and Paul Viola. Boosting image retrieval. *International Journal of Computer Vision*, 56(1-2):17–36, 2004.

[13] Zhuzhong Yang, Fangnian Lang, Xiaohong Yu, and Yu Zhang. The construction of fractional differential gradient operator. *Journal of Computational Information Systems*, 7(12):4328–4342, 2011.

Chapter 2

A Literature Review for Network Modeling and Analysis

2.1 Graph theory background

A *graph* is as an ordered pair of finite sets $G = (V, E)$. We refer to the elements of V as *vertices* or *nodes* and to the elements of E as *edges* or *links*. In its simplest variant, edges are formed by (undirected) pairs of vertices: $E \subseteq \{u, v \mid u \neq v \in E\}$ and so a graph captures some properties of the vertex set and, through the edges, of the pairwise relationships between the vertices. We say that an edge is *incident* on a vertex if the vertex belongs to the edge. For a given edge, we call all such vertices the *endpoints* of the edge.

If a simple graph does not capture all the information we require about a structure, additional types of information can be easily tacked on to this definition. The current definition is that of a *simple graph*. If E is allowed to have elements of the type uu, these are called *self-loops*. If we consider the elements of E as *ordered pairs*, that is, we distinguish between (u, v) and (v, u), we call G a *directed graph*. If E is allowed to be a *multiset*, we may call G a *multigraph* and the number of times $\mu(e)$ an edge e appears in E the *multiplicity* of that edge. There are situations were edges may be composed of more than a pair of vertices, and the resulting objects are called *hypergraphs*. We do not expect to study hypergraphs often in the scope of this work and so we will not go into any details here.

In order to fix other properties of an object represented by a graph, we may add various attributes to the vertices or edges. For example, an edge might have various numerical values associated with it. In a graph representing a road network, each edge might have a *length*. In a graph representing a pipeline, each edge might have both a *length* and *capacity*. In general, we may refer to a numerical value associated with each edge of a graph as a *weight*. In case of a directed graph, the values associated with (u, v) and with (v, u) may not be the same.

We define the *degree* of a vertex in a simple graph as the number of edges incident on the vertex: $d(v) = |\{e \in E \mid v \in e\}|$. The *neighborhood* of v $\Gamma(v)$ (sometimes denoted by $N(v)$) is the set of all *neighbors* of v: $\Gamma(v) = \{u \in V \mid uv \in E\}$. For a multigraph, we count each edge incident on the vertex with its multiplicity: $d(v) = |\{\sum_{e \in E \mid v \in e} \mu(e)\}|$. In a multigraph, the neighborhood of a vertex is a multiset. For a directed graph, we define the *outdegree* of a vertex as the number of

directed edges leaving the vertex: $d^-(v) = |\{v \in V \mid (u,v) \in E\}|$. Similarly, the *indegree* of a vertex as the number of directed edges entering the vertex: $d^+(v) = |\{v \in V \mid (v,u) \in E\}|$. In directed graphs, we must further specify whether we mean the *out-neighborhood* $\Gamma^+(u) = \{v \in V \mid (u,v) \in E\}$ or the *in-neighborhood* $\Gamma^-(v) = \{u \in V \mid (u,v) \in E\}$.

Analogous definitions apply to directed multigraphs. Similarly, in cases there is a weight associated with each edge, we sometimes define the (weighted) indegree (or outdegree) of a vertex as the sum of weights of the edges entering (or leaving) the vertex.

Given a graph $G = (V, E)$, a *subgraph* of G is any graph $F = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. Given a subset $U \subseteq V$, the *subgraph induced by U on G*, $G[U]$ is the subgraph of G with vertex-set U and the maximal edge set: $G[U] = (U, E')$ where $E' = \{uv \in E \mid \{u, v\} \subseteq U\}$.

There are several families of graphs with special structure that appear often enough to be worth defining. A *path* is a graph whose vertices can be arranged in order $v_0, v_1, \ldots v_k$ so that for every $i \leq k$, $v_{i-1}v_i \in E$ and every edge is of this form. Clearly, $d(v_0) = d(v_k) = 1$ and the degree of every other vertex is 2. We call v_0 and v_k the *endpoints* of the path. We say that there is a path from u to v in G if there exists a subgraph of G that is a path with endpoints u and v. A graph G is *connected* if for every pair of vertices x, $y \in V$, there is a path $v_0, \ldots, v_k$ in G such that $v_0 = x$ and $v_k = y$.

A *cycle* is a graph whose vertices can be arranged in order $v_0, v_1, \ldots v_k$ so that for every $i \leq k$, $v_{i-1}v_i \in E$, $v_kv_0 \in E$ and every edge is of this form. In a cycle, the degree of every vertex is exactly 2.

For directed graphs, the definitions of path and cycle may be strengthened by requiring that while walking along the edges of the path or cycle we only follow edges along their proper orientation. Sometimes this distinction is useful and we refer to a *directed path* or *directed cycle* as opposed to a path or cycle in the underlying undirected graph.

A *tree* is an undirected graph that is connected and has no cycles. It can be shown that in a tree, there exists a unique path between any two vertices. A vertex of degree 1 in a tree (and sometimes more generally in any graph) is called a *leaf*.

A *rooted tree* is a tree whose one vertex is specially designated as a *root*. A rooted tree is often visualized with the root at the top, and each edge from the root leading downwards. In a rooted tree with root r, it is easy to define *layers* $L_0, L_1, \ldots$ by setting $L_0 = \{r\}$, $L_i = \{v \in T \mid \ell(r,v) = i\}$ for $i \geq 0$, where $\ell(u,v)$ is the *length* of (the number of edges in) the unique $u - v$ path. For any vertex v in a rooted tree, we refer to the vertices along the unique path from the root to v (except for v itself) as v's *ancestors*. The vertex immediately preceding v on this path is the *parent* of v. Any vertices *below* v are its *descendants* and the neighbors of v immediately below v are its *children*. If u and v have a common parent, we say they are *siblings*. Given two vertices u, v in a tree T rooted at r, and the layers $L_0, L_1, \ldots$ defined as above, the *least common ancestor (LCA)* of u and v is the common ancestor w of u and v that lies in the maximum possible layer (furthest from the root; closest to u and v).

A *complete graph* is one that has all the possible edges. The notion of a complete subgraph is particularly useful, and has its own name: a *clique* in a graph G is any subgraph H of G such that $uv \in E$ for every u, $v \in V(H)$. Conversely, an *empty graph* is one that has no edges. (An empty subgraph is an empty graph which is an induced subgraph.)

We do not make a strict distinction between the terms *graph* and *network*, although we often refer to networks when additional information is present and we mostly talk about graphs when focusing on the basic combinatorial and topological structure.

2.1.1 Node similarity measures

We review some measures used to express similarity between a pair of nodes in a graph. Most of these are used in [17] as low-level ranking measures for link prediction, but many appear in other situations as well and so it may be useful to have them all in one place.

- *Graph/geodesic distance.* $d(x, y)$ = (Negated) length of the shortest path between x and y (when there is no danger of confusion, we use $d(x, y)$ as the notation for graph distance between x and y even though $d(v)$ still refers to the degree of the vertex v).
- *Number of common neighbors.* $|\Gamma(x) \cap \Gamma(y)|$.
- *Jaccard's coefficient.* $(\Gamma(x) \cap \Gamma(y))/(\Gamma(x) \cup \Gamma(y))$.
- *Adamic/Adar* [1]. $\sum_{z \in \Gamma(x) \cap \Gamma(y)} (\log(\Gamma(z)))^{-1}$.
- *Preferential attachment.* $|\Gamma(x)| \cdot |\Gamma(y)|$.
- *Katz$_\beta$.* $\sum_{\ell=1}^{\infty} \beta^\ell \cdot |P_{x,y}^{\langle \ell \rangle}|$, where $P_{x,y}^{\langle \ell \rangle} = \{\text{paths of length exactly } \ell \text{ from } x \text{ to } y\}$, and for $\ell = 1$, and in the unweighted version $P_{x,y}^{\langle 1 \rangle} = 1$ iff x and y collaborate.
- *Hitting time* or *Commute time.* Hitting time $H_{x,y}$ is defined as the expected number of steps a random walk on the graph starting from x takes to reach y. Either $-H_{x,y}$ or the normed version $-H_{x,y} \cdot \pi_y$ (where π_y is the probability of y in the stationary distribution) may be used. Since hitting time is not symmetric, the commute time $C_{x,y} = H_{x,y} + H_{y,x}$ may be used instead, or the normed version $H_{x,y} \cdot \pi_y + H_{y,x} \cdot \pi_x$.
- *Rooted PageRank$_\alpha$.* This is the stationary distribution probability of y under the random walk that at each step with probability $1 - \alpha$ follows a random edge and with probability α jumps back to the start node x.
- *SimRank* [11].

2.1.2 Graph measures

Here we list a few graph measures that have been used in characterizing networks and in some of the work we study later on network change detection. Additionally, some recent results on notions of connectivity and its generalizations follow.

- *Density.* The density of a graph is the proportion of the total possible number of edges that are present in the graph: $\rho(G) = |E|/(|V| \cdot (|V| - 1))$. Graphs with high density tend to be well connected and most of their vertices are close to each other. While this measure is suitable for characterizing graphs with unweighted links there is no clear single alternative if weighted links are present. Additionally, density should be viewed cautiously when comparing graphs with different numbers of nodes, e.g. a limit on the maximum degree of nodes implies larger graphs are less dense.
- *Betweenness centrality.* Betweenness centrality measures how often a vertex is on a shortest path between other vertices. Betweenness for the node k is defined as $b_k = \sum_{i,j} g_{ikj}/g_{ij}$, where g_{ikj} is the number of geodesic (shortest) paths between i and j that pass through k, and g_{ij} is the total number of shortest paths between i and j.

- *Closeness.* The closeness (global centrality) of a node is the inverse of the sum of its distances to all other nodes in the graph.
- *Eccentricity.* The eccentricity of a node is the maximum of the distances to all other nodes.

Batagelj and Zaveršnik [3] define a new notion connectivity in graphs. Two vertices $u, v \in V(G)$ are said to be k-gonally connected if there exists a sequence of cycles $C_1, \ldots, C_s$ with $|C_i| \leq k$ for $i = 1, \ldots, s$ such that $u \in C_1$, $v \in C_s$, and $C_i \cap C_{i+1} \neq \emptyset$. It is shown that this notion of connectivity defines an equivalence relation on the vertices of G. The authors refer to these equivalence classes as the k-gonal connectivity components. The authors generalize this notion of connectivity by replacing cycles (with cardinality bounds) with other classes of graphs.

In subsequent work Batagel and Zaveršnik [4] demonstrate an algorithm for computing the k-core decomposition of a graph having complexity $O(|E(G)|)$. They then generalize this algortihm to generalized cores (based on local vertex property functions) having complexity $O(m\dot{\max}(\Delta, \log(n)))$. Note that a local vertex property function is a vertex property function which only depends on the immediate neighors of a particular vertex, e.g. $\deg(v)$, $\text{in} - \text{degree}(v)$, $\text{out} - \text{degree}(v)$, etc. The algorithm makes use of a Fibonacci heap to efficiently maintain the most up-to-date vertex property function values for each $v \in V(G)$.

In this paper Bonchi et al [5] define an analogue of the k-core of a graph applicable in the case when the edges of the graph have some uncertainty. The model used for the graph assumes

$$P(e, f \in E(G)) = P(e \in E(G))P(f \in E(G)),$$

i.e. independence of the edges of G. The analogue of k-core that is introduced, namely the (k, η)-core for $k \in \mathbb{N}$ and $\eta \in [0, 1]$, is defined to be the maximal subgraph, say H, of G such that $P(\deg_H(v) \geq k) \geq \eta$ for all $v \in V(H)$. The authors describe a dynamic programming algorithm (and an enhancement with the same complexity but a better constant) which can compute the (k, η)-core decomposition of a graph (note that this decomposition is a hierarchy of (k, η)-cores for all $k \in \mathbb{N}$) in time $O(m\Delta)$ where $m = |E(G)|$ and Δ is the maximum degree of G. For comparison note that the k-core decomposition of a deterministic graph can be calculated in time $O(|E(G)|)$.

In this paper Feder and Mihail [7] define the notion of a balanced matroid. A balanced matroid is a matroid M such that $P(e \in B | f \in B) \leq P(e \in B)$ where B is a uniform random basis of M' where M' is any minor of M. One concrete class may be obtained by taking the bases to be the edges contained in the spanning trees of a graph, i.e.

$$B(G) = \{E(T) : T \text{ is a spanning tree of } G\}.$$

Note that a minor of a graphic matroid $M(G)$ is any $M(H)$ where H is any graph obtaineable from G via a series of edge deletions and contractions. It is mostly because of this class of matroids (known as graphic matroids) that this paper is of interest to us. The authors also show that a certain natural random walk (basis exchange) on the bases of a balanced matroid M is rapidly mixing. This has algorithmic consequences in terms of the complexity randomized algorithm for computing the approximate cardinality of the set of bases.

This paper is similar to the short cycle connectivity paper, except here the Saito et al [25] say two vertices are $u, v \in V(G)$ are "connected" if they are joined by a sequence of edges, say $e_1, \ldots, e_s$ such that e_i is contained in at least k triangles (3-cycles) for each $i = 1, \ldots, k$. Like the short cycle connectivity paper, this notion of connectivity defines an equivalance class on the vertices of the graph G. The authors discuss an algorithm for extracting the k-dense communities (i.e. the

equivalance classes.) They do not show any analysis of its complexity, but do mention that it should be closely related to that of computing clustering coefficients (which should be pretty reasonable.) The k-dense algorithm is compared with k-cores and with k-cliques on several datasets.

An edge $e \in E(G)$ is said to be most vital with respect to the number of spanning trees, if the number of spanning trees of G containing e is maximum (the most vital edge need not be unique.) This paper [29] discusses a $O(n^{2.376})$ for computing the most vital edge based on Kirchoff's matrix-tree algorithm. This result is mostly of interest because the fraction of spanning trees not passing through some edge is a monotonic edge property function. So the algorithmic result of this paper should be helpful in formulating an efficient implementation of the generalized core algorithm on edge where the edge property function is defined at $f(e)$ ="fraction of spanning trees of G not containing e."

2.1.3 Network structure and community extraction

Analyzing the structure of network beyond individual vertex or edge properties requires a more global approach. A particularly natural problem is that of *community identification*, where the goal is to partition the network into subsets of nodes such that the nodes within each subset are more closely related to each other than to the nodes in other subsets. The particular measure of strength of the relationship between nodes is open and in various applications may depend highly on properties other than the link structure, but if only link structure is considered, a useful notion of the strength of the community structure in the network was introduced by Newman and Girvan [20] and called *modularity*. Intuitively, given a partition of a network into groups, modularity is the number of edges within groups minus the expected number of edges in an equivalent network if the edges were placed at random. The approach of maximizing modularity has been successful in producing good network partitions in various examples of real and simulated social networks. Its generalizations include variants for directed networks by Leicht and Newman [15] and for detecting overlapping communities [2]. However, all of these require the whole network to be partitioned at once and it may not be a good idea to force every node into a community. Alternative approaches include the algorithm of Zhao et al [31] which extracts one community at a time by looking for a set with a large number of links within itself but a small number of links to the rest of the network. This algorithm can also be justified by showing that its estimated labels are asymptotically consistent if the network is generated by the stochastic block model.

2.2 Overview of main problems

The major problems related to networks that we address in this work can be broadly placed in the following three contexts:

1. Reconstructing a network from noisy link observations.

   In many scenarios the links are scores generated either manually or by an algorithm; each score representing the probability that an event linking the two endpoint locations happened during the period of observation. Since the actual network is unknown, missing and spurious links add uncertainty and may cause errors in the network structure. This may be alleviated using a network model to derive for each link a reliability value and based on these values either confirm the presence of a link, include a missing link (correct a false negative) or remove a spurious link (correct a false positive).

2. Network change detection

   Our applications come mostly from contexts in which networks evolve over time and our ultimate goal is to be able to detect changes in network structure as they happen. Thus, a large portion of our work will focus on the study of networks and graphs that change with time and on evaluating and detecting changes in such graphs. Among other things, this will require the development of models for time-evolving networks analogous to those used in noisy link network reconstruction.

   Most of the literature surveyed here addresses various versions of the change detection problem in time-evolving networks.

3. Network evolution and event modeling

   The final portion of our work will address the modeling of events that consist of more than a single change in the network composition or structure, for example complex events that are composed of a sequence of more primitive events. We will address this problem later.

2.3 Noisy link observations

In order to reconstruct the true network from a collection of noisy link observations, we must introduce some additional information beyond the observations themselves. In otherwords, we must make some (hopefully weak) assumptions on the structure of the network.

Liben-Nowell and Kleinberg [17] seem to be the first to ask the question "To what extent can the evolution of a social network be modeled using features *intrinsic to the network itself?*. Their most prominent sample dataset is based on the scientific collaboration data from which a co-authorship network can be derived. The nodes in this graph are individual researchers, and links are formed between two nodes if the two researchers appear as co-authors on a paper. Each link has the year of publication as an attribute, which makes this an early examples of a time-evolving social network. In particular, the question asked is, given two time intervals $[t_0, t_0']$ and $[t_1, t_1']$ and given the network $G[t_0, t_0']$ induced by exactly the edges present during the first time interval, output a list of edges not present in $G[t_0, t_0']$ that are predicted to appear in the network $G[t_1, t_1']$. Liben-Nowell and Kleinberg consider several methods for link prediction. The first set of methods are the low-level ones, each of which assigns a score to each potential link (pair of nodes) based on the input graph. The links can then be ranked by this score and an initial (deterministic or random) subset taken as the prediction. These low-level methods include graph distance, the number of common neighbors, Jaccard's coefficient, preferential attachment, hitting time, page rank and others. A second family consists of higher-level approaches that can be used in conjunction with any of the previous ones. These include the low-rank matrix approximation approach and clustering. The reported performance of all of the methods studied is relatively low, most likely because the setting in which they were evaluated is quite difficult: there are likely many factors outside of the link structure of the graph that influence new collaboration opportunities and so the input data cannot capture all the information required for prediction. However, this is the first attempt we are aware of a systematic approach to the link prediction problem in graphs.

Prior to [17], the only work on link prediction we are aware of focuses on settings where a lot more information is available than just the link structure of the graph. Popescul and Ungar [22] address the problem of predicting citations of research papers with the goal of designing an algorithm that recommends further papers to read given an abstract, author names and partial list of references.

Their approach relies not only on the link structure, but also on other available object attributes. A similar setting is considered by Taskar et al [28], who develop the *relational Markov network* framework for this purpose. For example, one of their scenarios involves predicting links between students using a social network based on the personal information (entered by students when creating their social network accounts) and a subset of links. Taskar et al report that using patterns based on node pairs and node triplets in building their model improves classification accuracy when compared to models based only on single node properties.

Clauset et al [6] define a hierarchical random graph model and use it to derive an algorithm for predicting missing links. The topological structure underlying this model is the *dendrogram*, a tree structure that results from a hierarchical clustering procedure. This is a rooted tree whose leaves are the graph vertices. Each non-leaf node A of the tree also has a value $p(A)$ associated with it, which is interpreted as the probability that two nodes with the least common ancestor A are linked by an edge. At a high level, their approach to link prediction is to generate an ensemble of dendrograms that fit the observed network and then look for pairs of vertices whose average probability of connection within this ensemble is high ("In general, we find that the top 1% of such predictions are highly accurate."). One of the networks considered in this paper is the network of associations among terrorists involved in the 9/11 attacks as derived by Krebs [13].

Stochastic block models were introduced by [30] in order to study the structure of groups of nodes within a network. The idea is to partition the node set into subsets of nodes that connect to the remainder of the network in a homogeneous way. In the simplest case, a block model is specified by a partition of nodes into groups and a matrix Q in which each element Q_{ij} represents the probability that a node in group i connects to a node in group j. The original paper [30] explicitly considers multiple networks simultaneously. In other words, each link may be labeled as belonging to one of several categories and multiple matrices $Q^1, \ldots, Q^k$ are used to specify the intergroup connection probabilities. For each of the link labels, a different matrix and thus a different partition may exist.

Guimerà and Sales-Pardo [9] use the stochastic block model to derive algorithms for reconstruction of networks whose links come from noisy observations. They show how to compute the reliability of a link, which is defined as the probability the link truly exists given an observation of the whole network (and the chosen family of stochastic block models) and then give criteria to detect both missing links and spurious links. They compute link reliability estimates by using a version of the Metropolis algorithm to sample the relevant node partitions. In order to go from individual link reliability calculations to the reconstruction of the full network, they further approximate *network reliability* $R_A^N = p_{BM}(A \mid A^O)$ (the probability that A is the true network given the observation A^O). This is again done using the Metropolis algorithm. Once network reliability can be computed, it is possible to calculate the expected value of any network property X as $\langle X \rangle = \sum_A X(A) R_A^N$, where the sum is over all possible networks. Since the number of terms in this summation is prohibitively large, $X(A^R)$ is used instead, where A^R is the network that maximizes R_A^N. The network A^R is referred to as the network reconstruction. Guimera and Sales-Pardo find that in many situations the reliability of A^R is much higher than that of the observed network A^O. This approach appears to outperform the hierarchical random graph approach of [6] in most tested examples.

2.4 Network evolution and change detection

Unlike the case of static networks, there has been much less work on detecting change in network structure over time. Most of the work in this area takes one or more network measures and detects change in each of them independently. For example, McCulloh and Carley [18, 19] study change in the structure of social networks with a focus on terrorist cells and a case study of Al Qaeda. Their approach is to use statistical process control to look for change points in a predefined measure of interest. The specific schemes they consider are the cumulative sum (CUSUM) [21], the exponentially moving average [23] and the scan statistic [8], but they mostly recommend CUSUM as the primary method for longitudinal network analysis and use closeness as the measure to which the CUSUM chart is applied.

Robinson and Priebe [24] study change-point and anomaly detection in network by using a new model of network change. In their model, links between pairs of nodes are established at random points in time (the application in question is email). Just as in the collaboration network example used by Liben-Nowell and Kleinberg, the links here have no duration: once established, they are considered present. A key feature of this approach is that the nodes are embedded in a low dimensional space and this embedding enables the representation of inhomogeneties in the connectivity of individual nodes and clustering of nodes. With each link, a categorical attribute is associated, that is, links come classified into one of K categories. In a particular application, these attributes may be observed directly, although in the example under study, an additional procedure is necessary to classify the links (emails from the Enron corpus) into categories. Robinson and Priebe propose a model in which the presence and attributes of observable edges depend (probabilistically) on the positions of their endpoints in a (latent) Euclidean space. The basic model is as follows. For each time t_i (taken from a Poisson process with rate λ), an edge opportunity is generated. To do this, a random pair of vertices u, v is selected from the population and their latent positions X_u and X_v generated randomly. Then with probability equal to the dot product $X_u \cdot X_v$, an edge is created. The latent positions are taken from a $(K+1)$-dimensional Dirichlet distribution.

Heard et al [10] present a two-stage method for anomaly detection in large dynamic networks. The first stage develops pairwise count models for assessing the normality of links, nodes, and aggregate traffic, while the second stage employs spectral analysis tools on a subset of the original candidate nodes. Importantly counts are treated as independent as such correlations between edges or nodes are not considered, greatly simplifying computation. In order to account for the sparsity in link observations a zero-inflated count process is constructed where by the presence of link is determined by the outcome of a Bernoulli process and if that link is present the value is then modeled as a Poisson, Markov Model, or via other non-parametric approaches. The construction is such that the overall increment in the counting process is accounted for by a compensator and the variations in the counting process martingale satisfy the standard criteria. Results are shown in both a sequential and complete (retrospective) data analysis. The focus was to develop a system which could detect changes in "real" time — in this case they were able to show detection capabilities in a "real" time scenario which improved upon previous analysis.

Snijders et al [26] present a continuous time Markov process on the space of relationships in a social network (unweighted digraph). While a novel model and inference framework are developed, a key contribution of this work is an efficient framework for the inference of interpretable factors which influence the development of the network. The continuous time Markov process is actor-based in the sense that the evolution of the graph is characterized by a sequence of time points at which an actor (node) has the opportunity to change one link. The advantage to such a mechanism

is that more closely approximates the dynamics of the evolution of networks and allows dependence to exist between consecutive states. The probability that an actor chooses to change a link is characterzied by an objective function which accounts for covariates enabling conclusions to be drawn with respect to the strength of influence, e.g. does gender impact the likelihood of link formation. Maximum likelihood estimates are obtained by leveraging the independence of activity sequences between consecutive observation times and data augmentation (integrating out times) to construct a sample path of changes. The results shown are similar to those obtained under Method of Moments estimation, but are relatively computationally expensive for even small data sets, i.e. $<$40 actors.

With the development of algorithms for streaming data sets, the idea of low-dimensional embedding of a complex data set has become popular in several fields of application. In our context, there is some interesting work on using sketch-type data structures to detect change in network data streams. The first such work is by Krishnamurti et al [14]. The actual change-point detection procedure used can be any used for univariate time series change detection (for example, a moving average, simple or shaped), but the difference is that by using a sketch-type data structure it may be possible to simultaneously consider a very large number of measures, even exponentially many. The trick is hidden in the sketch data structure. Originally developed for applications that require the distribution (that is, counts) of arbitrary keys in a data stream while using only a fixed amount of memory, a sketch data structure is basically a table whose entries are modified with each new stream element. Which entries of the table are modified is determined by hashing the new stream element through several randomly chosen hash functions. In Li et al [16], this is even clearer: they explicitly take random projections of vectors whose values describe network flows observed at a particular moment and apply univariate change detection to each of the components of the resulting vector.

Sun et al [27] give an algorithm that attempts to compress the graph in order to estimate its complexity. The change detection step then reduces to detecting change in the compressed size. The heuristic used for compression is based on sequentially partitioning each graph snapshot into communities, and then comparing the relative cost of adding that partition to the previously compressed graph sequence to the cost of starting a new separate compression subsequence. The community detection procedure is an iterative partitioning heuristic that attempts to partition each graph in the sequence into subgraphs that are either almost complete or almost empty. This is done by considering the graph as a bipartite graph (or directed graph formed by splitting each vertex into an in-vertex and an out-vertex half) and then alternating between the in-vertex side and the out-vertex side, rearranging the partition to maximize an entropy measure. It is interesting to compare the change points found by this procedure to those found by [24].

A few recent works study the change in network community structure by extracting a sequence of community sets, either independently each time or, more interestingly, by taking into account the previously detected community structure and weighing the observed network change against the change in previously detected communities [12]. This weighing is already present in an implicit form in [27] because of the way the compression heuristic works with the graph sequence.

REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] O. Bakun. *Adaptive Decentralized Routing and Detection of Overlapping Communities*. PhD thesis, ASU, 2011.

[3] V. Batagelj and M. Zaveršnik. Short cycle connectivity. *Discrete Mathematics*, 307(3):310–318, 2007.

[4] V. Batagelj and M. Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5(2):129–145, 2011.

[5] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1316–1325. ACM, 2014.

[6] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 05 2008.

[7] T. Feder and M. Mihail. Balanced matroids. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 26–38. ACM, 1992.

[8] R. A. FISHER, H. G. THORNTON, and W. A. MACKENZIE. The accuracy of the plating method of estimating the density of bacterial populations. *Annals of Applied Biology*, 9(3-4):325–359, 1922.

[9] R. GuimerÃă and M. Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078, 2009.

[10] N. A. Heard, D. J. Weston, K. Platanioti, D. J. Hand, et al. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645–662, 2010.

[11] G. Jeh and J. Widom. Mining the space of graph properties. In *KDD*, pages 187–196, 2004.

[12] V. Kawadia and S. Sreenivasan. Sequential detection of temporal communities by estrangement confinement. *Sci. Rep.*, 2, 11 2012.

[13] V. Krebs. Mapping networks of terrorist cells. *CONNECTIONS*, 24(3):43–52, 2002.

[14] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: methods, evaluation, and applications. In *Internet Measurement Comference*, pages 234–247. ACM, 2003.

[15] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100:118703, Mar 2008.

[16] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement 2006, Rio de Janeriro, Brazil, October 25-27, 2006*, pages 147–152, 2006.

[17] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[18] I. McCulloh. *Detecting Changes in a Dynamic Social Network*. PhD thesis, Carnegie Mellon University, 2009.

[19] I. McCulloh and K. M. Carley. Detecting change in longitudinal social networks. *Journal of Social Structure*, 12, 2011.

[20] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb. 2004.

[21] E. S. Page. Cumulative sum charts. *Technometrics*, 3(1):1–9, 1961.

[22] A. Popescul, L. H. Ungar, S. Lawrence, and D. M. Pennock. Statistical relational learning for document mining. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, pages 275–282, 2003.

[23] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1(3):239–250, 1959.

[24] L. F. Robinson and C. E. Priebe. Detecting Time-dependent Structure in Network Data via a New Class of Latent Process Models. *ArXiv e-prints*, Dec. 2012.

[25] K. Saito, T. Yamada, and K. Kazama. Extracting communities from complex networks by the k-dense method. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(11):3304–3311, 2008.

[26] T. A. B. Snijders, J. Koskinen, and M. Schweinberger. Maximum likelihood estimation for social network dynamics. *Ann. Appl. Stat.*, 4(2):567–588, 06 2010.

[27] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In P. Berkhin, R. Caruana, and X. Wu, editors, *KDD*, pages 687–696. ACM, 2007.

[28] B. Taskar, M.-f. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *in Neural Information Processing Systems*, 2003.

[29] F.-S. P. Tsen, T.-Y. Sung, M.-Y. Lin, L.-H. Hsu, and W. Myrvold. Finding the most vital edges with respect to the number of spanning trees. *Reliability, IEEE Transactions on*, 43(4):600–603, 1994.

[30] H. White, S. Boorman, and R. Breiger. Social structure from multiple networks: I. blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–80, 1976.

[31] Y. Zhao, E. Levina, and J. Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.